An Efficient Load Balancing Technology in CDN

YUN BAI[1], BO JIA[2], JIXIANG ZHANG[3], QIANGGUO PU[1], NIKOS MASTORAKIS[4]
[1]College of Information and Electronic Engineering,
University of Science and Technology of Suzhou, Jiangsu 215011, CHINA
[2]College of Information and Electronic Engineering,
Zhejiang Gongshang University, Zhejiang 310018, CHINA
[3]Institute of Graphics and Image, Hangzhou Electronic University, Zhejiang 310018, CHINA
[4]Hellenic Naval Academy, TermaHatzikyriakou, 18539 Piraeus, GREECE
Also: WSEAS, Ag. I. Theologou17-23, 15773, Zografou, Athens, GREECE

Abstract: - The existing load balancing technology in CDN only emphasizes even load distribution among servers, and it doesn't make full use of network topology information and file access history, so user request can't get timely response. To solve the above problem, this paper puts forward a load balancing algorithm based on the Distributed binning strategy. This algorithm can make full use of network topology information and file access history as well as server load information, analyzing the popularity of files with the access history of the clusters of clients from the servers, efficiently finishing distribution and routing of the high popularity files among servers so that users can closely obtain the required contents, ease Internet congestion and enhance response speed of user accessing websites.

Key-Words: - CDN, distributed binning strategy; history record, load balancing, network topology

1 Introduction

At present, the technology of the content distribution network(CDN) has been extensively applied in Internet. Using CDN technology can make a user rapidly gain requested content. It efficiently eases network congestion and increases response speed of the user access to network [1-3]. The load balancing technology is one of the four elements in CDN [4]. It can ensure that the user request points to the nearest edge server with minimum load in the network, so that network content is efficiently distributed.

The conventional load balancing technology in CDN mainly emphasizes a server load evenness index, and ignores the efficient application of network topology structure and file access history, which leads to long response time to a user request, even causes request failure[5]. As for each user request, CDN requests the center server to timely construct the network topology map of this request so as to measure the network distance, so the center server becomes the bottleneck of the whole network. If many user requests occur at the same time, the performance of the center server will fall rapidly, and ultimately affect running of the whole network. To efficiently realize load balancing, a certain strategy must be adopted to timely gain network topology information. People with K. W. Ross as the representative propose adopting IP address of network node to generate topology information [6]. Since there are many false IP addresses in the network, authenticity of topology information is not good. People with B. Krishnamurthy as the representative propose extracting topology information from BGP table of Internet core router [7]. Since some network structures are not managed by BGP, there is no BGP table in the core router, so it can't gain all topology information. People with Ratnasamy as the representative proposes adopting Distributed binning strategy to get topology information, and use ICMP ping method to measure RTT values from each node to some datum landmark node, and group gained RTT into vector in order, and then use this vector to represent topology information of network node[8]. After researching, people with J. Chu as the representative find out that 10% file account for 60% flow of the whole network, which shows that there is a big difference in file access frequency(that is, "hotness")[9]. Based on disadvantage and limitation of current method, we put forward a load balancing algorithm in CDN based on the Distributed binning strategy. This algorithm can make full use of network topology information and file access history as well as server load information, analyzing "hotness" of a file and efficiently finishing the content distribution and routing of a "hot" file.

2 Load balancing technology

2.1 Distributed binning strategy

To efficiently realize load balancing in CDN, first of all, there are two problems to be solved. One is that how to gain and maintain dynamic network topology information; and the other one is that how to find the edge server close to user group. Adopting the Distributed binning strategy is the best method of rapidly gaining edge server and topology information of client terminal. Requiring less about network infrastructure, this strategy is practical and extendable. The network topology information provided by applying this strategy can efficiently solve the problem of load balancing in CDN.

Distributed binning strategy:

1. Assume that Internet is equipped with some nodes called landmark, and other network node can gain IP addresses of these landmark nodes by checking DNS.

2. By using ICMP ping, each node measures the RTT from itself to each landmark so as to get a node table concerning "Distributed binning" of relative distances from network nodes to these landmarks, and arrange RTT according to sort ascending.

3. As for time delay to different landmark, each node has a corresponding time delay in sort ascending. This sort represents the bin that this node belongs to, so that adjacent nodes have similar sort information, that is, adjacent nodes in network topology belong to the same bin.

4. After gaining bin of the node, make quantified handling with time delay of landmark. Divide time delay into 3 levels: level 0, time delay is 0~100ms; level 1, time delay is 101~200ms; level 2, time delay is more than 200ms. After quantified handling, it is known that the relative distance from network node to each landmark has a corresponding level. For instance, in Figure 1, three landmark nodes are used, the distance from node A to landmark L1, L2, L3 is 230ms, 55ms, and 115ms respectively. Therefore, the sort of relative distance from node A to each landmark node is L2L3L1, level vector of node A is "0 1 2", and bin of node A is "L2L3L1:012".

In this strategy, one network node only needs to measure the distance between itself and the landmark, and there is no need to know the distance between landmarks, and there is also no need to gain from the landmark node the distance of other node to the landmark. The landmark node doesn't need to make cross measurement, nor collect and expand the information measured. Distributed binning is very healthy to the fault with one or more landmark nodes.

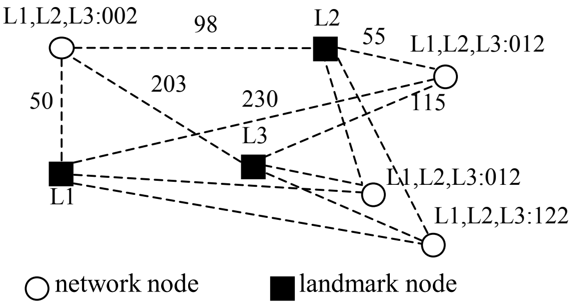

Fig. 1 Distributed Binning illustration

2.2 Network model for load balancing

Based on the above-mentioned Distributed binning strategy, assume that each edge server and each client terminal in CDN have their corresponding labels: IP address. With such topology information, we can easily realize load balancing in CDN about client clusters, edge server choices and overall dynamic.

To describe the load balancing in CDN, firstly, we need to construct a network model. Assume that there are N network servers in the network, and that the capacity of these servers is C_1, C_2, …, C_n respectively, the responsible zone is Z_1, Z_2, …, Z_n respectively, and network load is W_1, W_2, …, W_n respectively. There are M files distributed in the network. Access cost of these files is O_1, O_2, … , O_m respectively, and the corresponding access frequency is f_1, f_2, …, f_m. In ideal load balancing, the network load distributed to each server should be in direct ratio to service capacity of server itself. Therefore, a server G should meet the equation (1).

$$\frac{W_G}{\sum_{j=1}^{N} W_j} = \frac{C_G}{\sum_{j=1}^{N} C_j} \qquad (1)$$

Definition 1: $W_K = \sum_{x=\alpha_1}^{\alpha_e} f_x O_x$, where, W_K means total load of server K in CDN, f_x means access frequency of file x stored in the server, O_x means access cost of file x stored in the server, the file stored in the server includes α_1, α_2, …, α_e.

In practice, such perfect network model for load balancing is impossible. The goal of our load balancing is to provide optimal load balancing technology for CDN in small area. The following algorithm for content distribution and content routing based on the Distributed Binning strategy can realize this goal.

2.3 Algorithm for content distribution

The algorithm for content distribution needs synthetically to take “hotness” degree of a file, effect of file size on server load, network distance between edge server and requesting client, etc. into comprehensive account. We establish a file access history table in CDN server, using $C_i(T)$ to stand for it, where subscript i means index number of a file. $C_i(T)$ means the access times of user group from the same sort vector T in the past time counted. The structure of the information table is shown in Figure 2. This table records the access of each file in the server in the past period of time. From this, we can analyze “hotness” degree of each file, and find out that which user group has highest demand for this file. The algorithm is summarized as follows:

1. For “hotness” of a file, set appropriate threshold value α .

2. When the times of accessing a file by user group with same sort vector exceed α , start duplicating a file, select an edge server from edge servers with this sort vector to store duplicate of a file. The selected server shall be adjacent to user group with high demand for this file in topology.

3. When next time a member of this user group requests this file, the requested file content (duplicate) can be obtained from the near edge server by the following routing algorithm. As shown in Figure 2, the file with index number 12 has 30 accesses from the user group with sort vector 121. In this way, handling with access user is made. Next, select one of edge server that has the same sort vector with this user group, and store duplicate of file in it. After, when the user with same sort vector requests this file, the center server will directly point the request to the edge server having the duplicate of this file.

4. For the convenience of selecting appropriate edge server to store the duplicate of a “hot” file, firstly, the server having this “hot” file contacts the edge server to store the duplicate by sending message. After receiving the message, the edge server responds with its own load W, making use of file access history, and calculate W by the formula in Definition 1. According to the response information, select the edge server with minimum working load to save the duplicate of this file. Finally, establish the direct link to the duplicate of this file in the center server.

5. When the center server finds that user T of the edge server R requests this file, it will timely forward the request to R.

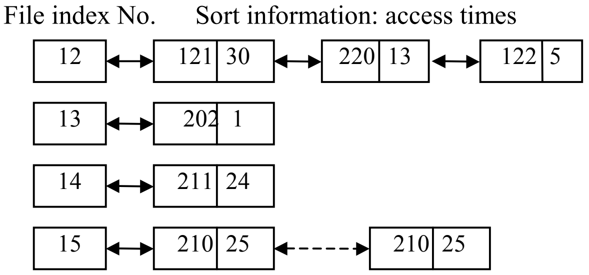

Fig. 2 Sample history information table on server A_{IP}

This algorithm efficiently makes use of network topology information and file access history obtained by the Distributed binning strategy, makes a reasonable analysis on “hotness” of a file according to file access frequency, and quickly selects the edge server adjacent to the requesting client cluster according to network topology information, and in the meantime, selects the edge server to store duplicate of the file, make use of the previously mentioned network model for load balancing and sufficiently measures the load condition of candidate servers.

2.4 Algorithm for content routing

The algorithm for content routing is to select optimal edge server by the network topology information obtained based on the Distributed binning strategy. In the meantime, it takes working load of each server into comprehensive account. It needs to periodically update the network topology information. This algorithm is summarized as follows:

1. For server group $F(f_1,f_2,\ldots,f_k)$ with calculated level vector and a client terminal C. Assume that the level vector corresponding to the server group F is （v_1，v_2，…,v_k）, and the level vector of the client terminal making request is $v_{c.}$

2. Select a server f_x whose level vector v_x is same or similar as v_c from level vector of server group F.

3. If there are many such servers, calculate load sever W according to the Definition 1, and select a server with minimum load to return to client C, so as to meet equation (1) as far as possible. If there isn’t, return to the address of original content server.

3 Test result and analysis

Using C language and NS-2 network simulation software, we make a simulation test on the above algorithm for content routing. In the case of 20 servers, we compare two algorithms. User's network access time Tt is divided into 3 parts, that is, Tt=Tc+Tp+Tr, where, Tt is the time from user requesting to receiving response, Tc is the time from user requesting to server receiving the request, Tp is the handling time of the center server from receiving request to response leaving the center server, Tr is the time from response leaving the center server to user receiving response. For Simple algorithm and the algorithm based on Distributed Binning, under same network environment condition, Tc is equal to Tr. The main difference in user's network access time lies in handling time Tp of the center server with user request. Assuming Tc and Tr are constants, we use 4 landmark nodes in the simulation.

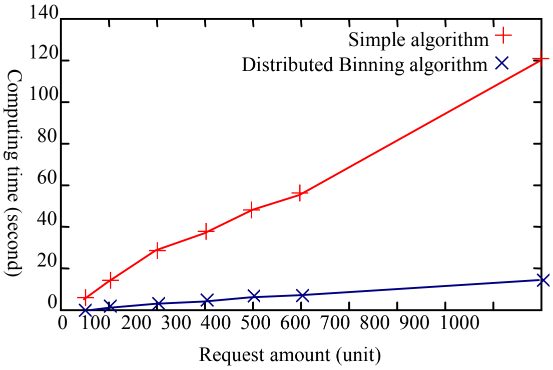

Fig. 3 comparing with Simple algorithm about 20 servers

When the number of server is 20, 50 respectively, we make simulation test on the performances of sample algorithm and the algorithm introduced in this paper. The test results are shown in Figure 3 and Figure 4. From Figure 3 and Figure 4, we know that comparing with Simple algorithm, the algorithm for content routing based on Distributed Binning remarkably reduces the handling time of the center server with the user request. With increase of number of server, the advantage of this algorithm is more superior. In Simple algorithm, for the divided server group and the host computer at client terminal that makes requests, the distance between the host computer at client terminal and each server in the network tree needs to be measured, and then select the edge server with minimum distance to return to requesting user. In the algorithm for content routing introduced in this paper, it doesn't need to measure network distance from client terminal to each edge server to select an optimal server. Only comparing the level vector provided by itself with the level vector that each server belongs to, can get the nearest server to the requesting user, and while considering the nearest network distance, it fully considers the load condition of the selected server, thus reduces the handling time of the center server, optimizes the network performance of CDN and improve application rate of Internet bandwidth.

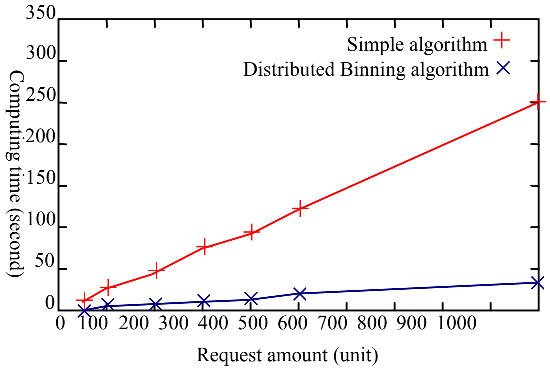

Fig. 4 comparing with Simple algorithm about 50 servers

4 Conclusions

This paper introduces a load balancing technology in CDN based on the Distributed Binning strategy, gives out a network model to measure load condition of a server, and proposes new algorithms for content distribution and content routing. This load balancing technology can efficiently finish content distribution and routing in CDN so that a user network request can get optimal response service, especially when there are more edge servers. It can greatly reduce time delay of network access, thus efficiently ease Internet congestion. This technology is helpful in researching more efficient models for measuring server load, so it has better application foreground.

References:

[1] Stardust. Com Inc., The Ins and Outs of Content Delivery Networks, White Papers, December 20,2000, http://www.digitalpipe.com/pdf/dp/whitepapers/cdn-streaming/cdn.pdf.

[2] Stardust. Com Inc., Content Networking and Edge Services : Leveraging the Internet for Profit, Stardust, Sept 19,2001 http://www.speakerforum.com/Uploads/Stardust/pdfs/cdn.whitepaper.pdf .

[3] Matthew L., Content Delivery Network (CDN) 2A Reference Guide, detail, April 10, 2001, http://Informationweek.com/detail/RES/9959901 81212.html.

[4] OLNN.CN.CDN, The Content distributed network technology, server, May 28, 2007, http://olnn.cn/html/server/6/20070528/3660.html

[5] W. D. Zhao, Content routing algorithms achieving network proximity in content delivery networks, Journal of Zhejiang University, Vol. 38, No. 4, p.414 -419, April 2004.

[6] B. Krishnamurthy, J. Wang, Y. Xie, Early Measurements of a Cluster-based Architecture for P2P System", ACM SIGCOMM, Nov. 2001.

[7] B. Krishnamurthy, J. Wang, On Network Aware Clustering of Web Clients, in ACM SIGCOMM, 2000.

[8] S. Ratnasamy, M. Handley, R. M. Karp and S. Shenker, Topologically-Aware Overlay Construction and Server Selection, INFOCOM, 2002.

[9] J. Chu, K. Labonte, B. N. Levine, Availability and Locality Measurements of Peer-to-Peer File Systems, in Proceedings of IT Com, Scalability and Traffic Control in IP Networks II Conferences, July 2002.